ECE 2020

# Stock price prediction using Convolutional Neural Network

# *Member*

1. Thịnh - Leader
2. Nam - Data collection
3. Hoàng - Model design & training
4. Trí - Made Powerpoint

# *Contents of the Report*

# *Introduction*


## What is stock?

Stock is a small fraction of a company. Stocks can be traded using money.

For example, FPT Telecom company stock cost 80600VND per stock.

Stock price can go up and down

→ Gain huge potential if we can predict

→ Use Convolutional Neural Network

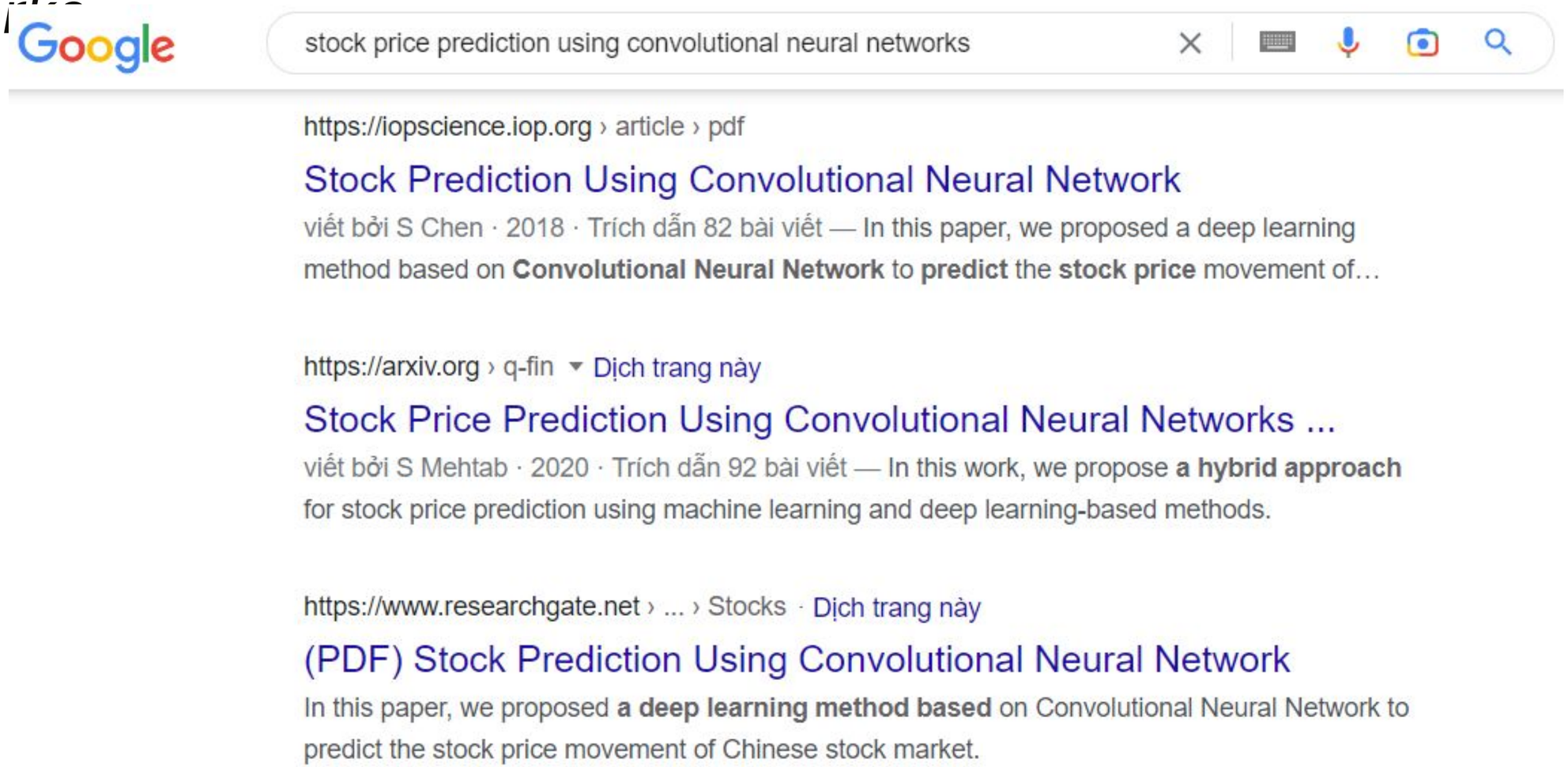
*Target*

**1**

Produce a model that can predict FPT stock price of the next day by using close price of the last 50 days.

**2** Analysze its accuracy.

# *Related works*



# *Data Preparation - 6 steps*

## *Step 1*


| A | B | C | D | E | F |
|---|---|---|---|---|---|
| Ticker>,<DTYYYYMMDD>,<OpenFixed>,<HighFixed>,<LowFixe | | | | | |
| FPT,20061213,15.6977,15.6977,15.6977,15.6977,83530 | | | | | |
| FPT,20061214,16.4826,16.4826,16.4826,16.4826,280710 | | | | | |
| FPT,20061215,17.3067,17.3067,17.3067,17.3067,265300 | | | | | |
| FPT,20061218,18.1701,18.1701,18.1701,18.1701,215790 | | | | | |
| FPT,20061219,19.0727,19.0727,19.0727,19.0727,137520 | | | | | |
| FPT,20061220,20.0146,20.0146,20.0146,20.0146,220120 | | | | | |
| FPT,20061221,20.0146,20.0146,19.0335,19.0335,156400 | | | | | |
| FPT,20061222,18.0916,18.0916,18.0916,18.0916,143500 | | | | | |
| FPT,20061225,17.1890,17.1890,17.1890,17.1890,22140 | | | | | |
| FPT,20061226,18.0131,18.0131,16.3649,18.0131,300150 | | | | | |
| FPT,20061227,18.0131,18.8373,18.0131,18.8373,413230 | | | | | |
| FPT,20061228,18.8373,18.8373,18.4449,18.8373,144840 | | | | | |
| FPT,20061229,18.2486,18.2486,18.0524,18.0524,69760 | | | | | |
| FPT,20070102,17.2675,17.6600,17.2675,17.2675,81170 | | | | | |
| FPT,20070103,17.6599,17.6599,17.2675,17.6599,74660 | | | | | |
| FPT,20070104,18.5233,18.5233,18.5233,18.5233,148510 | | | | | |
| FPT,20070105,19.4259,19.4259,19.4259,19.4259,377800 | | | | | |
| FPT,20070108,20.0146,20.0146,19.2297,19.2297,351470 | | | | | |
| FPT,20070109,19.4259,19.4259,19.2297,19.2297,225270 | | | | | |
| FPT,20070110,19.2297,19.4259,19.2297,19.4259,314250 | | | | | |
| FPT,20070111,19.6221,19.6221,19.6221,19.6221,283920 | | | | | |
| FPT,20070112,20.6032,20.6033,20.6032,20.6033,279710 | | | | | |
| FPT,20070115,20.9957,21.6237,20.9957,21.6236,274290 | | | | | |
| FPT,20070116,22.6832,22.6832,22.6832,22.6832,339760 | | | | | |
| FPT,20070117,23.5073,23.5073,21.5844,21.5844,145870 | | | | | |
| FPT,20070118,20.5248,21.6236,20.5248,21.6236,282430 | | | | | |
| FPT,20070119,22.6832,22.6832,22.6832,22.6832,95290 | | | | | |
| FPT,20070122,23.7820,23.7820,23.5466,23.5466,461630 | | | | | |
| FPT,20070123,23.5466,23.6251,23.5466,23.6251,243990 | | | | | |

Download the Financial Statement of FPT.

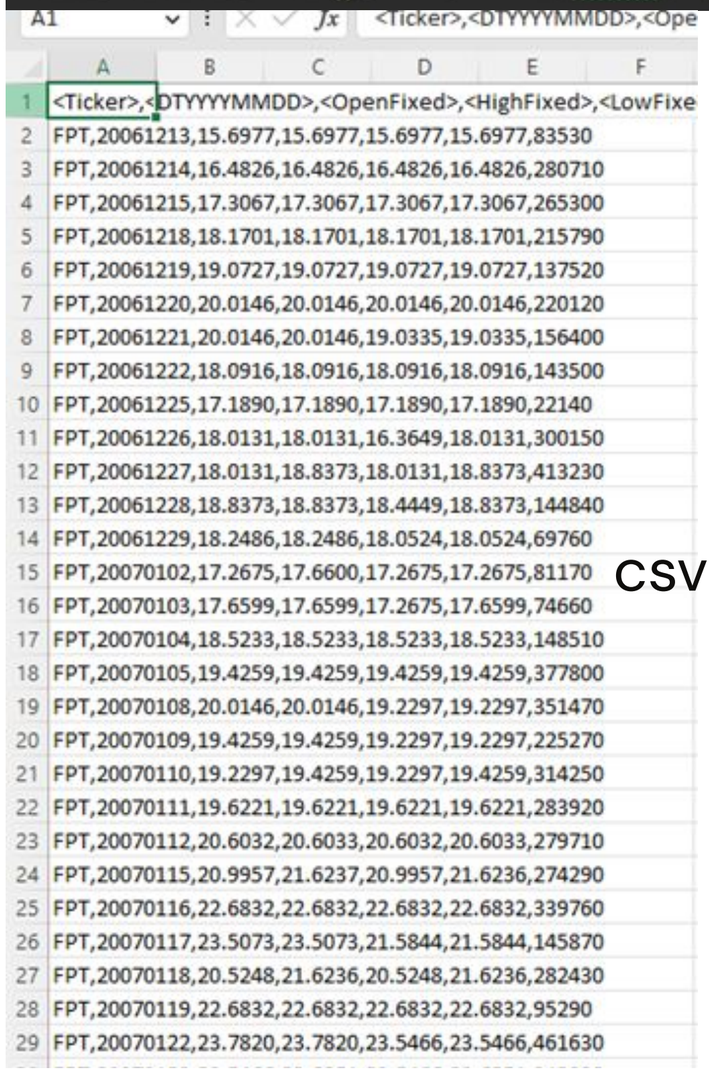
https://www.cophieu68.vn/export.php

# *Step 2*

The file was downloaded in.csv format; therefore, we must use a module called pandas to convert it to.xlsx format (regular Excel).

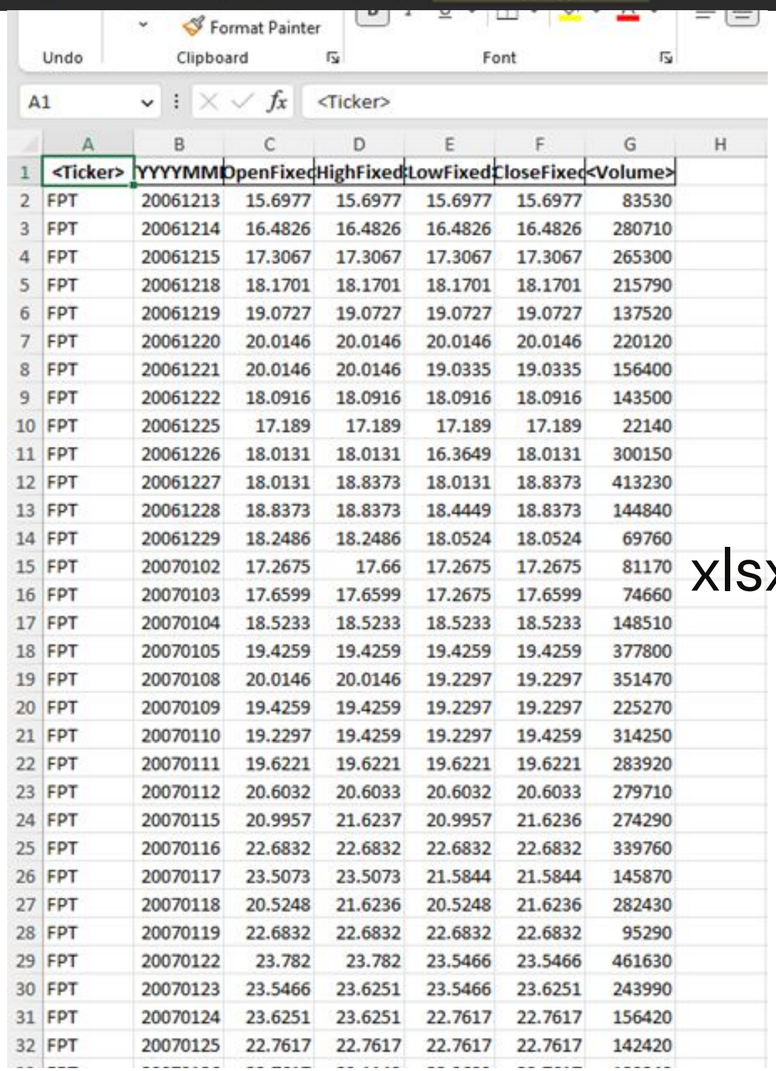
```
import pandas as pd

read_file = pd.read_csv (r'C:/Users/luciu/PycharmProjects/MCT1/fpt.csv')
read_file.to_excel (r'C:/Users/luciu/PycharmProjects/MCT1/fpt.xlsx', index = None, header=True)
```



| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | <Ticker>,<DTYYYYMMDD>,<OpenFixed>,<HighFixed>,<LowFixe | | | | | |
| 2 | FPT,20061213,15.6977,15.6977,15.6977,15.6977,83530 | | | | | |
| 3 | FPT,20061214,16.4826,16.4826,16.4826,16.4826,280710 | | | | | |
| 4 | FPT,20061215,17.3067,17.3067,17.3067,17.3067,265300 | | | | | |
| 5 | FPT,20061218,18.1701,18.1701,18.1701,18.1701,215790 | | | | | |
| 6 | FPT,20061219,19.0727,19.0727,19.0727,19.0727,137520 | | | | | |
| 7 | FPT,20061220,20.0146,20.0146,20.0146,20.0146,220120 | | | | | |
| 8 | FPT,20061221,20.0146,20.0146,19.0335,19.0335,156400 | | | | | |
| 9 | FPT,20061222,18.0916,18.0916,18.0916,18.0916,143500 | | | | | |
| 10 | FPT,20061225,17.1890,17.1890,17.1890,17.1890,22140 | | | | | |
| 11 | FPT,20061226,18.0131,18.0131,16.3649,18.0131,300150 | | | | | |
| 12 | FPT,20061227,18.0131,18.8373,18.0131,18.8373,413230 | | | | | |
| 13 | FPT,20061228,18.8373,18.8373,18.4449,18.8373,144840 | | | | | |
| 14 | FPT,20061229,18.2486,18.2486,18.0524,18.0524,69760 | | | | | |
| 15 | FPT,20070102,17.2675,17.6600,17.2675,17.2675,81170 | | | | | |
| 16 | FPT,20070103,17.6599,17.6599,17.2675,17.6599,74660 | | | | | |
| 17 | FPT,20070104,18.5233,18.5233,18.5233,18.5233,148510 | | | | | |
| 18 | FPT,20070105,19.4259,19.4259,19.4259,19.4259,377800 | | | | | |
| 19 | FPT,20070108,20.0146,20.0146,19.2297,19.2297,351470 | | | | | |
| 20 | FPT,20070109,19.4259,19.4259,19.2297,19.2297,225270 | | | | | |
| 21 | FPT,20070110,19.2297,19.4259,19.2297,19.4259,314250 | | | | | |
| 22 | FPT,20070111,19.6221,19.6221,19.6221,19.6221,283920 | | | | | |
| 23 | FPT,20070112,20.6032,20.6033,20.6032,20.6033,279710 | | | | | |
| 24 | FPT,20070115,20.9957,21.6237,20.9957,21.6236,274290 | | | | | |
| 25 | FPT,20070116,22.6832,22.6832,22.6832,22.6832,339760 | | | | | |
| 26 | FPT,20070117,23.5073,23.5073,21.5844,21.5844,145870 | | | | | |
| 27 | FPT,20070118,20.5248,21.6236,20.5248,21.6236,282430 | | | | | |
| 28 | FPT,20070119,22.6832,22.6832,22.6832,22.6832,95290 | | | | | |
| 29 | FPT,20070122,23.7820,23.7820,23.5466,23.5466,461630 | | | | | |

csv file



| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | <Ticker> | YYYYMMD | OpenFixed | HighFixed | LowFixed | CloseFixed | <Volume> | |
| 2 | FPT | 20061213 | 15.6977 | 15.6977 | 15.6977 | 15.6977 | 83530 | |
| 3 | FPT | 20061214 | 16.4826 | 16.4826 | 16.4826 | 16.4826 | 280710 | |
| 4 | FPT | 20061215 | 17.3067 | 17.3067 | 17.3067 | 17.3067 | 265300 | |
| 5 | FPT | 20061218 | 18.1701 | 18.1701 | 18.1701 | 18.1701 | 215790 | |
| 6 | FPT | 20061219 | 19.0727 | 19.0727 | 19.0727 | 19.0727 | 137520 | |
| 7 | FPT | 20061220 | 20.0146 | 20.0146 | 20.0146 | 20.0146 | 220120 | |
| 8 | FPT | 20061221 | 20.0146 | 20.0146 | 19.0335 | 19.0335 | 156400 | |
| 9 | FPT | 20061222 | 18.0916 | 18.0916 | 18.0916 | 18.0916 | 143500 | |
| 10 | FPT | 20061225 | 17.189 | 17.189 | 17.189 | 17.189 | 22140 | |
| 11 | FPT | 20061226 | 18.0131 | 18.0131 | 16.3649 | 18.0131 | 300150 | |
| 12 | FPT | 20061227 | 18.0131 | 18.8373 | 18.0131 | 18.8373 | 413230 | |
| 13 | FPT | 20061228 | 18.8373 | 18.8373 | 18.4449 | 18.8373 | 144840 | |
| 14 | FPT | 20061229 | 18.2486 | 18.2486 | 18.0524 | 18.0524 | 69760 | |
| 15 | FPT | 20070102 | 17.2675 | 17.66 | 17.2675 | 17.2675 | 81170 | |
| 16 | FPT | 20070103 | 17.6599 | 17.6599 | 17.2675 | 17.6599 | 74660 | |
| 17 | FPT | 20070104 | 18.5233 | 18.5233 | 18.5233 | 18.5233 | 148510 | |
| 18 | FPT | 20070105 | 19.4259 | 19.4259 | 19.4259 | 19.4259 | 377800 | |
| 19 | FPT | 20070108 | 20.0146 | 20.0146 | 19.2297 | 19.2297 | 351470 | |
| 20 | FPT | 20070109 | 19.4259 | 19.4259 | 19.2297 | 19.2297 | 225270 | |
| 21 | FPT | 20070110 | 19.2297 | 19.4259 | 19.2297 | 19.4259 | 314250 | |
| 22 | FPT | 20070111 | 19.6221 | 19.6221 | 19.6221 | 19.6221 | 283920 | |
| 23 | FPT | 20070112 | 20.6032 | 20.6033 | 20.6032 | 20.6033 | 279710 | |
| 24 | FPT | 20070115 | 20.9957 | 21.6237 | 20.9957 | 21.6236 | 274290 | |
| 25 | FPT | 20070116 | 22.6832 | 22.6832 | 22.6832 | 22.6832 | 339760 | |
| 26 | FPT | 20070117 | 23.5073 | 23.5073 | 21.5844 | 21.5844 | 145870 | |
| 27 | FPT | 20070118 | 20.5248 | 21.6236 | 20.5248 | 21.6236 | 282430 | |
| 28 | FPT | 20070119 | 22.6832 | 22.6832 | 22.6832 | 22.6832 | 95290 | |
| 29 | FPT | 20070122 | 23.782 | 23.782 | 23.5466 | 23.5466 | 461630 | |
| 30 | FPT | 20070123 | 23.5466 | 23.6251 | 23.5466 | 23.6251 | 243990 | |
| 31 | FPT | 20070124 | 23.6251 | 23.6251 | 22.7617 | 22.7617 | 156420 | |
| 32 | FPT | 20070125 | 22.7617 | 22.7617 | 22.7617 | 22.7617 | 142420 | |

xlsx file

## *Step 3*

Aftet that, we create another excel file that contains the data that we want to train, here is the closing price

| | | |
|---|---|---|
| 7 | 20170110 | 16.1421 |
| 8 | 20170111 | 16.1772 |
| 9 | 20170112 | 16.1246 |
| 10 | 20170113 | 15.7561 |
| 11 | 20170116 | 15.4403 |
| 12 | 20170117 | 15.5806 |
| 13 | 20170118 | 15.3701 |
| 14 | 20170119 | 15.3701 |
| 15 | 20170120 | 15.5105 |
| 16 | 20170123 | 15.5982 |
| 17 | 20170124 | 15.6859 |
| 18 | 20170125 | 15.8263 |
| 19 | 20170202 | 15.7561 |

## *Step 4*

Because we want to use the sliding window approach to construct the dataset.

For example, the data will be compiled:

[Day 1] to [Day 50] and [Day 51] is increase/decrease

[Day 2] to [Day 51] and [Day 52] is increase/decrease

...

[Day 250] to [Day 299] and [Day 300] is increase/decrease

We shall put 1 at the end of the row if the day after increases; else, we

```
15.9316,16.0719,16.1421,16.1421,16.3351,16.0895,16.0368,15.9491,16.0193,16.1246,16.0719,16.0719,15.9667,16.1421,16.0544,16.0368,15.9842,16.0368,16.107,16.3527,-1
15.9316,16.0719,16.1421,16.1421,16.3351,16.0895,16.0368,15.9491,16.0193,16.1246,16.0719,16.0719,15.9667,16.1421,16.0544,16.0368,15.9842,16.0368,16.107,16.3527,-1
16.0719,16.1421,16.1421,16.3351,16.0895,16.0368,15.9491,16.0193,16.1246,16.0719,16.0719,15.9667,16.1421,16.0544,16.0368,15.9842,16.0368,16.107,16.3527,16.3527,-1
16.1421,16.1421,16.3351,16.0895,16.0368,15.9491,16.0193,16.1246,16.0719,16.0719,15.9667,16.1421,16.0544,16.0368,15.9842,16.0368,16.107,16.3527,16.3527,16.1772,1
16.1421,16.3351,16.0895,16.0368,15.9491,16.0193,16.1246,16.0719,16.0719,15.9667,16.1421,16.0544,16.0368,15.9842,16.0368,16.107,16.3527,16.3527,16.1772,16.8439,-1
```

*Step 5*

**01**

After that, we import the openpyxl module to use an excel file.

```
import openpyxl as op
```

**02**

Then we set some basic data of the excel file.

```
wb = op.load_workbook("s.xlsx")
sh = wb.active
col = 2
num_rows = 40050
start = 1
row = 0
row_i = 2
count = 1
file = open("C:/Users/luciu/PycharmProjects/MCT1/s.txt", 'w')
```

**03**

When we write in a text file, the letter "w" at the end indicates that we are overwriting the text.

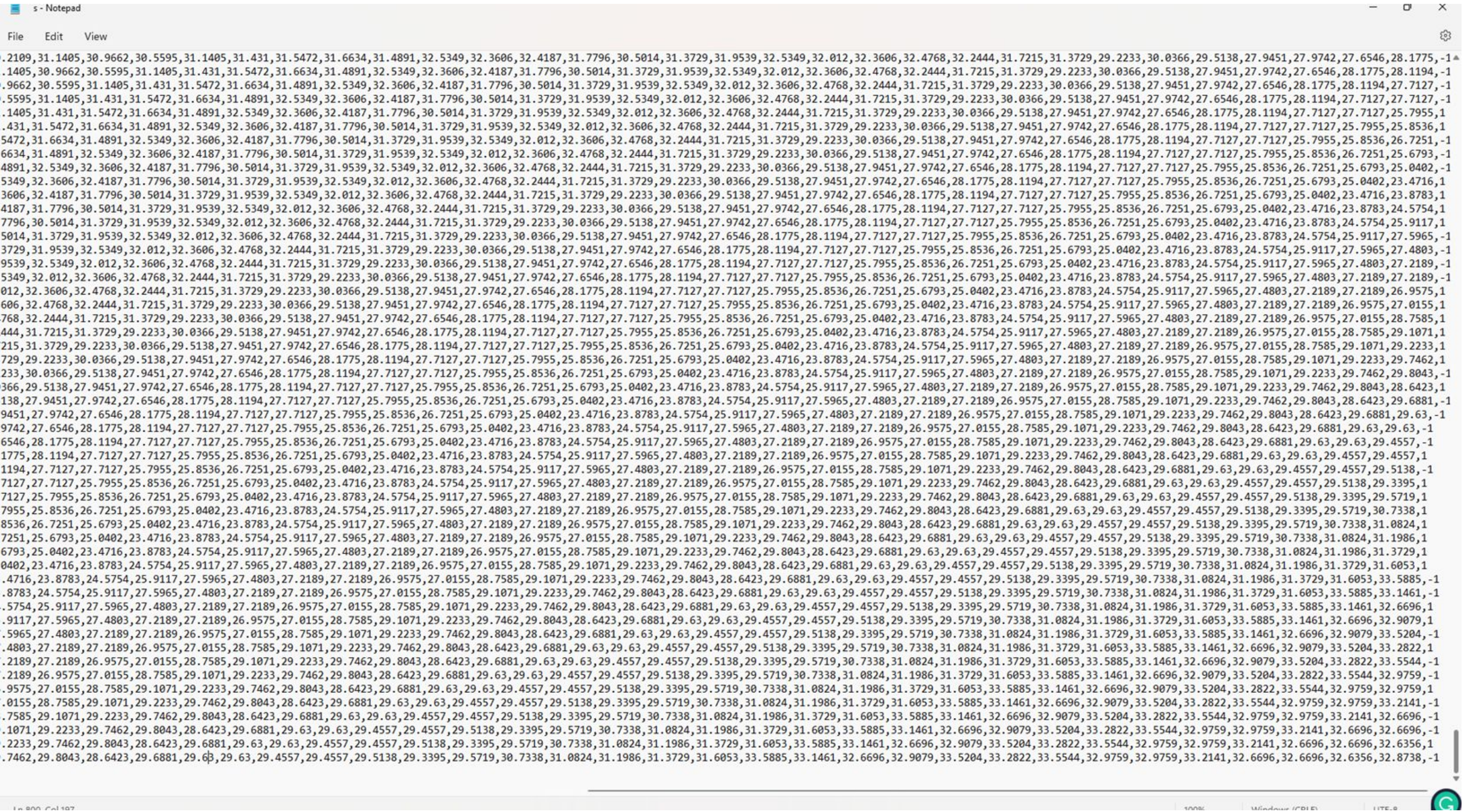
```
file = open("C:/Users/luciu/PycharmProjects/MCT1/s.txt", 'w')
```

# Step 6

Then, using a loop to separate each value from the others using commas, we write each value into the text file. When there are 50 values, it will compare the number from the previous day with the next day, and if the difference is greater, it will write -1; otherwise, it will write 1. The next line's starting value will be the value for the following day as we restart the counting. If there are not enough values to make 50, it will keep counting.

```
while row < num_rows+1 :
    file.write(str(sh.cell(row_i, col).value)+ ",")
    print(row)
    if count == 50:
        if sh.cell(row_i, col).value >= sh.cell(row_i +1, col).value:
            file.write('-1')
        else:
            file.write('1')

        start = start + 1
        row_i = start
        count = 1
        file.write('\n')
    else:
        row_i +=1
        count +=1
    row += 1
```




The result

# *Model training*

```
[1]  from google.colab import drive
     import pandas as pd
     import numpy as np
```

```
[2]  drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

**Connect to GG drive**

```
[3]  train=pd.read_csv(r'/content/gdrive/MyDrive/SSA_lab/train.txt', header=None)
     test=pd.read_csv(r'/content/gdrive/MyDrive/SSA_lab/test.txt', header=None)
```

[4] train

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 41 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 15.7386 | 15.8789 | 15.8263 | 15.7737 | 15.8614 | 16.1421 | 16.1772 | 16.1246 | 15.7561 | 15.4403 | ... | 16.0719 |
| **1** | 15.8789 | 15.8263 | 15.7737 | 15.8614 | 16.1421 | 16.1772 | 16.1246 | 15.7561 | 15.4403 | 15.5806 | ... | 15.9667 |
| **2** | 15.8263 | 15.7737 | 15.8614 | 16.1421 | 16.1772 | 16.1246 | 15.7561 | 15.4403 | 15.5806 | 15.3701 | ... | 16.1421 |
| **3** | 15.7737 | 15.8614 | 16.1421 | 16.1772 | 16.1246 | 15.7561 | 15.4403 | 15.5806 | 15.3701 | 15.3701 | ... | 16.0544 |
| **4** | 15.8614 | 16.1421 | 16.1772 | 16.1246 | 15.7561 | 15.4403 | 15.5806 | 15.3701 | 15.3701 | 15.5105 | ... | 16.0368 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **795** | 28.1775 | 28.1194 | 27.7127 | 27.7127 | 25.7955 | 25.8536 | 26.7251 | 25.6793 | 25.0402 | 23.4716 | ... | 32.9079 |
| **796** | 28.1194 | 27.7127 | 27.7127 | 25.7955 | 25.8536 | 26.7251 | 25.6793 | 25.0402 | 23.4716 | 23.8783 | ... | 33.5204 |
| **797** | 27.7127 | 27.7127 | 25.7955 | 25.8536 | 26.7251 | 25.6793 | 25.0402 | 23.4716 | 23.8783 | 24.5754 | ... | 33.2822 |
| **798** | 27.7127 | 25.7955 | 25.8536 | 26.7251 | 25.6793 | 25.0402 | 23.4716 | 23.8783 | 24.5754 | 25.9117 | ... | 33.5544 |
| **799** | 25.7955 | 25.8536 | 26.7251 | 25.6793 | 25.0402 | 23.4716 | 23.8783 | 24.5754 | 25.9117 | 27.5965 | ... | 32.9759 |

800 rows × 51 columns

[5] test

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 41 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 33.5885 | 33.3162 | 33.2141 | 31.3084 | 31.9890 | 31.3084 | 31.7168 | 31.5466 | 31.7168 | 31.9890 | ... | 31.6827 |
| **1** | 33.3162 | 33.2141 | 31.3084 | 31.9890 | 31.3084 | 31.7168 | 31.5466 | 31.7168 | 31.9890 | 31.9890 | ... | 31.9890 |
| **2** | 33.2141 | 31.3084 | 31.9890 | 31.3084 | 31.7168 | 31.5466 | 31.7168 | 31.9890 | 31.9890 | 32.3634 | ... | 31.8529 |
| **3** | 31.3084 | 31.9890 | 31.3084 | 31.7168 | 31.5466 | 31.7168 | 31.9890 | 31.9890 | 32.3634 | 32.1251 | ... | 32.0571 |
| **4** | 31.9890 | 31.3084 | 31.7168 | 31.5466 | 31.7168 | 31.9890 | 31.9890 | 32.3634 | 32.1251 | 31.8529 | ... | 32.1251 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **195** | 56.1708 | 56.3098 | 55.4756 | 54.7109 | 54.7804 | 53.5291 | 53.2510 | 52.9034 | 53.9462 | 53.5986 | ... | 60.4809 |
| **196** | 56.3098 | 55.4756 | 54.7109 | 54.7804 | 53.5291 | 53.2510 | 52.9034 | 53.9462 | 53.5986 | 54.0157 | ... | 63.1226 |
| **197** | 55.4756 | 54.7109 | 54.7804 | 53.5291 | 53.2510 | 52.9034 | 53.9462 | 53.5986 | 54.0157 | 55.4061 | ... | 66.3900 |
| **198** | 54.7109 | 54.7804 | 53.5291 | 53.2510 | 52.9034 | 53.9462 | 53.5986 | 54.0157 | 55.4061 | 55.4756 | ... | 65.2081 |
| **199** | 54.7804 | 53.5291 | 53.2510 | 52.9034 | 53.9462 | 53.5986 | 54.0157 | 55.4061 | 55.4756 | 55.3365 | ... | 64.9996 |

200 rows × 51 columns

# Importing dataset to Python

```
[6]  xtrain=train.iloc[:,0:50]
     xtrain=np.array(xtrain)
     xtrain=np.reshape(xtrain,(800,50))
```

```
[7]  ytrain=train.iloc[:,50:51]
     ytrain=np.array(ytrain)
     ytrain=np.reshape(ytrain,(800,1))
```

```
[8]  xtest=test.iloc[:,0:50]
     xtest=np.array(xtest)
     xtest=np.reshape(xtest,(200,50))
```

```
[9]  ytest=test.iloc[:,50:51]
     ytest=np.array(ytest)
     ytest=np.reshape(ytest,(200,1))
```

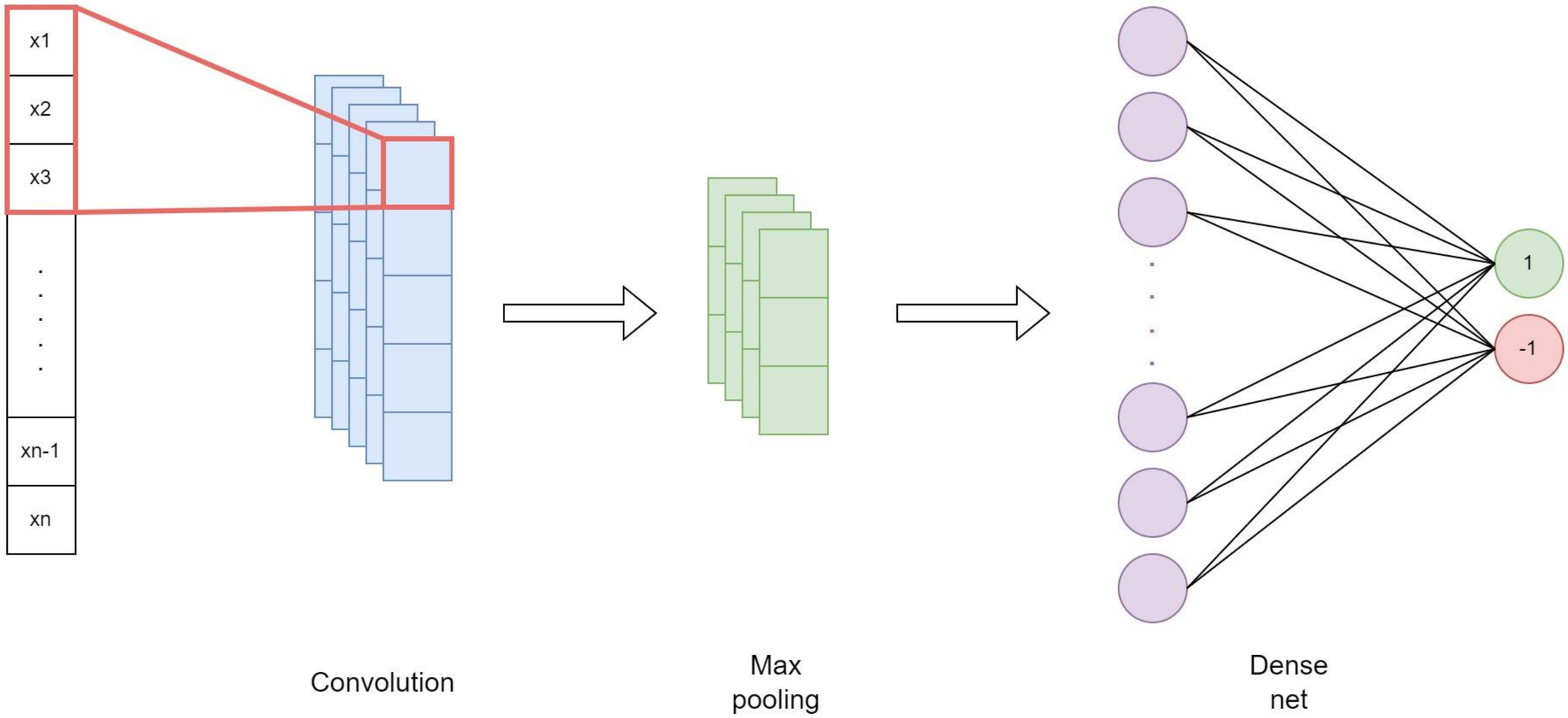
# Training process

```
model = keras.models.Sequential([
    keras.layers.Conv1D(filters = 32,
                                kernel_size = 2,
                                input_shape = (50,1),
                                activation = "relu"),
    keras.layers.MaxPooling1D(2),
    keras.layers.Flatten(),
    keras.layers.Dense(2, activation='softmax')
    ])

model.compile(optimizer='adam', loss='mse', metrics=['accuracy'])
model.fit(xtrain, ytrain, epochs=1000)
```

```
Epoch 973/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2755 - accuracy: 0.2375
Epoch 974/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2729 - accuracy: 0.2387
Epoch 975/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2729 - accuracy: 0.2325
Epoch 976/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2732 - accuracy: 0.2400
Epoch 977/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2724 - accuracy: 0.2438
Epoch 978/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2748 - accuracy: 0.2675
Epoch 979/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2742 - accuracy: 0.2525
Epoch 980/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2730 - accuracy: 0.2587
Epoch 981/1000
25/25 [==============================] - 0s 3ms/step - loss: 1.2725 - accuracy: 0.2488
Epoch 982/1000
```

# Model summary



## Giving prediction

```
[[0.6592454  0.34075463]]
```

## Evaluate the accuracy

```
[11] test_loss = model.evaluate(xtest, ytest)

     7/7 [==============================] - 0s 4ms/step - loss: 1.1591 - accuracy: 0.3700
```

# *Future development*

-Collect more data

-Use more features (Open price, volume,...)

-Add more layers into the model

-Use different techniques (LSTM, Random forest,...)

-Try on other stocks (VCB, VIN, ACB,...)

# Thank you!

Ask us if u have any question.